LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

1

LA-UR -86-3887

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

LA-UR--86-3887

DE87 002902

TITLE: FAST PARALLEL ALGORITHMS FOR THE MOORE-PENROSE PSEUDO-INVERSE

AUTHOR(S): Maurice W. Benson
Paul O. Frederickson

SUBMITTED TO: Second Conference on Hypercube Multiprocessors
Knoxville, Tennessee, September 29 - October 1, 1986

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO 836 R4

Fast Parallel Algorithms for the Moore-Penrose Pseudo-Inverse

Maurice W. Benson*
Paul O. Frederickson**

Abstract.

We employ the concept of an approximate pseudo-inverse (API) of a singular linear operator A to construct algorithms that yield the Moore-Penrose pseudo-inverse solution $x^+ = A^+y$ to the singular system $Ax = y$. Such an algorithm is a nonmatrix representation of the Moore-Penrose pseudo-inverse A^+ of A. We are particularly interested in fast algorithms [executable in $O(n)$ operations] that are fully efficient on parallel computers. One result presented concerns the construction of APIs for consistent problems and is applied to the two-dimensional free-boundary spline interpolation problem. For this problem, the conjugate gradient algorithm, preconditioned with an API, proves to be very effective. The V-cycle multigrid algorithm FAPIN is shown to be an API if the smoother satisfies a certain condition and proves to be very effective for Poisson's problem on a two-torus. We have demonstrated our algorithms for n close to one million on the iPSC hypercube at Christian Michelsen Institute, Bergen.

1. Introduction

If the large sparse linear operator $A: \mathcal{X} \rightarrow \mathcal{Y}$ is singular, then the linear system

$$Ax = y \tag{1}$$

may have no solution, which is the case when $y \notin \mathcal{R}(A)$. In this situation, our task is often to find a *best approximate solution*, or a *least squares fit* to the data y, which is any x such that $\| y - Ax\|$ is minimized. When $\mathcal{X}$ and $\mathcal{Y}$ are finite dimensional Hilbert spaces, our primary concern in numerical computation, not only do such best approximate solutions

* *Department of Mathematical Sciences, Lakehead University, Thunder Bay, Canada.*

** *Los Alamos National Laboratory, Los Alamos, New Mexico.*

x always exist, but there is a unique best approximate solution of minimum norm. This we denote x^+ and refer to as the Moore-Penrose solution of equation (1). The operator $A^+ : Y \to X : y \mapsto x^+$ turns out to be linear and is called the Moore-Penrose pseudo-inverse of A. For more details see Moore [14], Penrose [15], or Ben-Israel and Greville [1].

Our task is to derive fast parallel algorithms for A^+, algorithms which will allow us to evaluate $x^+ = A^+y$ as efficiently as possible on a parallel computer. Our approach is to define an *approximate* pseudo-inverse to A and show that these can sometimes be constructed efficiently on a parallel computer. When they can, they provide an effective construction of the exact Moore-Penrose pseudo-inverse A^+.

Definition 1. *The linear operator $Z: \mathcal{Y} \to \mathcal{X}$ is an ϵ-approximate pseudo-inverse (ϵ–API) of the linear operator $A: \mathcal{X} \to \mathcal{Y}$ if $\epsilon < 1$ and*

$$\| (Z - ZAZ)v\| \le \epsilon \, \| Zv \| \quad \forall v \in \mathcal{Y} , \qquad \mathcal{N}(Z) \perp \mathcal{R}(A) , \ \textit{and} \ \mathcal{R}(Z) \perp \mathcal{N}(A) . \tag{2}$$

Using the concept of an ϵ–API, we can state a simple defect correction (DC) algorithm for the solution of (1) as the following:

algorithm API-DC

$$r^n = y - Ax^n, \quad x^{n+1} = x^n + Zr^n$$

Theorem 1. *If Z is an $\epsilon-$ approximate pseudo-inverse of A, then algorithm API-DC converges at the geometric rate ϵ, for any initial approximation x^0, to an x such that $\| y - Ax\|$ is minimized. If $x^0 = 0$, then x is the Moore-Penrose pseudo-inverse solution $x^+ = A^+y$.*

A detailed proof may be found in Frederickson and Benson [9]. For a fuller discussion of the iterative solution of singular systems, see Ben-Israel and Greville [1] and Keller [12]. Our primary interest in this paper is to show that there are highly parallel algorithms which implement the concept of an API and therefore, through Theorem 1, provide a fast representation of A^+. In the next section, we show that the orthogonality condition in $\mathcal{X}$ is handled easily for consistent systems. In Section 3, we show that a preconditioned conjugate gradient algorithm can be an effective representation of A^+ if an API is used as a preconditioner. Phrased another way, CG-acceleration of the API–DC algorithm is possible under certain conditions. We demonstrate this on a rather large singular problem, one of dimension 500,000 and with a null space of dimension 3000, on an iPSC hypercube. For details see Section 5. In Section 4, we show that the multigrid algorithm FAPIN (fast approximate pseudo-inverse) can provide an effective representation of an API on a hypercube.

2. Construction of API's in the Consistent Case

Consistent, but underdetermined, linear systems are particularly easy to handle because we are able to concentrate on the inequality in the definition of API and ignore the orthogonality conditions.

Theorem 2. *If the linear operator $A : \mathcal{X} \rightarrow \mathcal{Y}$ is onto, that is $\mathcal{R}(A) = \mathcal{Y}$, and the linear operator $B : \mathcal{Y} \rightarrow \mathcal{X}$ satisfies the inequality*

$$\| x - BAx \| \leq (\sqrt{1+\epsilon} - 1) \| x \| \qquad \forall x \perp \mathcal{N}(A) \tag{3}$$

for some $\epsilon < 1$, then $Z = A^* B^* B$ is an $\epsilon - API$ of A.

Proof: $\mathcal{R}(Z) \perp \mathcal{N}(A)$ follows from the fact that $\mathcal{R}(Z) = \mathcal{R}(A^*)$. For any $x \perp \mathcal{N}(A)$

$$\| (I - A^* B^* BA)x \| = \| (I - (I - (I - A^* B^*))(I - (I - BA)))x \|$$

$$\leq \| (I - A^* B^*)x \| + \| (I - BA)x \| + \| (I - A^* B^*)(I - BA)x \| \leq \epsilon \| x \|.$$

Since $\mathcal{R}(A) = \mathcal{Y}$, $\mathcal{N}(Z) \subset \mathcal{R}(A)$. Thus if $\mathcal{N}(Z) \neq 0$, we can find $x \in \mathcal{X}$ such that $ZAx = 0$, violating the inequality. ∎

As demonstrated on the spline interpolation problem below, the above approach can provide an effective solution method. For related results, see Björck and Elfving[4] or Elfving [6], where a symmetric successive overrelaxation technique is used as a preconditioner for the conjugate gradient method applied to $AA^* x = y$.

3. Approximate Inverses

Suppose now that the linear spaces $\mathcal{X}$ and $\mathcal{Y}$ have bases $\{x^i,\ i \in I\}$ and $\{y^j,\ j \in J\}$, with respect to which the linear operator A has a sparse matrix representation. In some cases, we are able to construct an API that also has a sparse matrix representation, perhaps with the same sparsity as A^*. One way to do this generalizes the concept of an LSQ approximate inverse [2,3] that has proven quite effective for nonsingular linear systems.

Definition 2. *The LSQ approximate pseudo-inverse of A is the operator $Z = \Pi_X B \Pi_Y$, where Π_X projects onto $\mathcal{N}^{\perp}(A)$, Π_Y projects onto $\mathcal{R}(A)$, and B minimizes the Frobenius norm*

$$\| I - BA \|_F \tag{4}$$

subject to the constraint that B have the same graph (non-zero structure) as A^*.

For example, the bicubic spline interpolation operator on a two-torus is defined by the 9-point operator

$$A = \frac{1}{36} \begin{pmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{pmatrix} , \tag{5}$$

which is both sparse and diagonally dominant. An elementary calculation shows that

$$Z = LSQ(A) = \begin{pmatrix} 0.1541 & -0.6603 & 0.1541 \\ -0.6603 & 2.830 & -0.6603 \\ 0.1541 & 0.6603 & 0.1541 \end{pmatrix} . \tag{6}$$

Note that Z has the same cartesian product structure as A. Even in the variable coefficient case, this construction is inexpensive and parallel because the variational equations that

conditions.

for some interesting problems (such as the Laplacian on a two-torus, which we consider in Section 4), the projections $\Pi_{\mathcal{X}}$ and $\Pi_{\mathcal{Y}}$ are inexpensive. In other cases, the method suggested by Theorem 2, which does not rely on the availability of efficient projection operators, is more appropriate. This is what we use to precondition the conjugate gradient operator in the next section.

4. API-CG: An API-Preconditioned Conjugate Gradient Algorithm

If the operator $A: \mathcal{X} \rightarrow \mathcal{Y}$ is nonsymmetric as well as singular, we can be sure that the standard conjugate gradient algorithm will have difficulty. The recent papers of Faber and Manteuffel [7] and [8] contain a clear discussion of the limitations of the standard conjugate gradient algorithm and the class of generalizations that they refer to as orthogonal error methods. On the other hand, Kammerer and Nashed [11] and Björck and Elfving [4] and [6] have shown that the preconditioned conjugate gradient algorithm of Concus, Golub, and O'Leary [5] may be used, under certain conditions, to solve singular linear systems. We will see that when Z is an API of A with the additional property that $ZA: \mathcal{X} \rightarrow \mathcal{X}$ is symmetric, the following preconditioned conjugate gradient algorithm converges.

algorithm API-CG

$$
\begin{aligned}
\textit{initiate}: \quad & r^0 = y - Ax^0 \\
& q^0 = Zr^0 \\
& p^0 = q^0 \\
\textit{iterate}: \quad & a^i = \langle q^i, r^i \rangle / \langle p^i, Ap^i \rangle \\
& x^{i+1} = x^i + a_i p^i \\
& r^{i+1} = r^i - a_i A p^i \\
& q^{i+1} = Z r^{i+1} \\
& b_i = \langle q^{i+1}, r^{i+1} \rangle / \langle q^i, r^i \rangle \\
& p^{i+1} = q^{i+1} + b_i p^i
\end{aligned}
$$

We have obtained particularly good convergence rates using the inner product $\langle x, y \rangle = (x, A^+ y)$ in conjunction with an API of the form $Z = A^* B^* B$. The evaluation of both inner products required by the API-CG algorithm is easy in this situation. To understand this, note that

$$\langle q^i, r^j \rangle = (Zr^i, A^+ r^j) = (r^i, B^* B A A^+ r^j) = (Br^i, Br^j), \tag{7}$$

with the last reduction using the fact that $r^j \in \mathcal{R}(A)$. Observing that $p^j \in \mathcal{R}(Z) = \mathcal{R}(A^*)$, we compute

$$\langle p^i, Ap^j \rangle = (p^i, A^+ Ap^j) = (p^i, p^j), \tag{8}$$

which is certainly easy to evaluate.

Theorem 3. *If Z is an $\epsilon-$ approximate pseudo-inverse of A and $ZA: \mathcal{X} \to \mathcal{Y}$ is symmetric, then the API-CG algorithm converges, for any $y \in \mathcal{Y}$ and any $x^0 \in \mathcal{X}$, to an x such that $\| y - Ax\|$ is minimized. The error at the n^{th} iterate satisfies the inequality*

$$\| x - x^n \| < 4(\epsilon/2)^{2n} \| x^0 \| . \tag{9}$$

If $x^0 = 0$, then the limit x is the Moore-Penrose pseudo-inverse solution $x^+ = A^+ y$. Thus the API-CG algorithm with $x^0 = 0$ is an effective representation of the Moore-Penrose pseudo-inverse A^+ of A.

For a proof we refer the reader to Reference [9].

5. The Fast Approximate Pseudo-Inverse FAPIN

If the linear operator $A : \mathcal{X} \to \mathcal{Y}$ is very poorly conditioned as well as singular, we may expect it difficult to find an ϵ-API that is powerful enough to make even the API–CG algorithm converge as fast as we would like. We do, however, know how to deal with discretizations of elliptic partial differential equations: the antithesis of their ill-conditioning is the multigrid algorithm in one of its many variants. We describe one V-cycle of a multigrid algorithm as an approximate inverse. If the smoothing is done properly, using an API of A, and if a projection is inserted in the right place, the result is an approximate pseudo-inverse, which we refer to as a *fast approximate pseudo-inverse*, or FAPIN. Choosing the correct smoothing step is the key to building this algorithm.

Definition 3. *The linear operators $Z_k: \mathcal{Y}_k \to \mathcal{X}_k$ are an ϵ-nested approximate pseudo-inverse (ϵ-NAPI) of the linear operators $A_k: \mathcal{X}_k \to \mathcal{Y}_k$ if $\epsilon < 1$ and*

$$\| (I - Z_k A_k) u \|^2 \le \epsilon^2 \| u' \|^2 + \| u'' \|^2 \quad \forall u \perp \mathcal{N}(X_k) ,$$

$$\mathcal{N}(Z_k) \perp \mathcal{R}(A_k) , \;\; \textit{and} \;\; \mathcal{R}(Z_k) \perp \mathcal{N}(A_k) ,$$

where $u = u' + u''$, $u' \in \mathcal{N}(A_{k-1})$, $u'' \perp \mathcal{N}(A_{k-1})$.

algorithm FAPIN(k, r_k **)**

if $k > 0$, *then*

$$r_k = \Pi_k r_k$$
$$r_{k-1} = P_k r_k$$
$$w_{k-1} = FAPIN(k-1, r_{k-1})$$
$$w_k = Q_k w_{k-1}$$
$$s_k = r_k - A_k w_k$$
$$w_k = w_k + Z_k s_k$$

else

$$w_k = Z_0 r_0$$

endif

return w_k

Theorem 4. *If for some $\epsilon < 1$ and all $0 \leq j \leq k$, Z_j is a nested approximate pseudo-inverse of A_j, then $FAPIN(k, *)$ is an ϵ–API of A_k, and may be used in either API–DC or API–CG to provide an effective representation of the Moore-Penrose pseudo-inverse A^+ of A.*

The interested reader is referred to Reference [9] for a detailed proof.

5. Hypercube Computations

We study a free boundary spline interpolation problem in the plane. Consider a discrete mesh with spacing h in the x and y directions and with n_x and n_y points in the x and y directions, respectively. At each *interior* point, a value for the function to be interpolated is provided. Thus, applying the bicubic spline interpolation operator at these points yields $(n_x - 2)(n_y - 2)$ equations in $n_x n_y$ unknowns, a consistent system for any data because the operator is of full rank.

We have implemented the algorithms presented here on the 32-node iPSC hypercube at Christian Michelsen Institute in Bergen, Norway. We used the high-level CMI library as the basis for our computations and added an efficient procedure that evaluated any of the operators A, A^*, B, and B^*, using a 7 by 7 array of 3 by 3 arrays to store the coefficients. The API–CG algorithm requires more communication cost than the API-DC algorithm because of the inner products. As shown in Table I, this is more than offset by the enhanced performance.

Table I. Computational Results for the Free-Boundary Spline Problem*

		API–DC			API–CG	
N	$per(1)$	$per(8)$	$T(8)/N$	$per(1)$	$per(8)$	$T(8)/N$
2^{11}	0.332	0.271e-3	4.69	0.125	0.732e-5	5.88
2^{13}	0.341	0.321e-3	3.36	0.123	0.754e-5	4.02
2^{15}	0.344	0.325e-3	2.98	0.119	0.716e-5	3.50
2^{17}	0.346	0.328e-3	2.86	0.120	0.674e-5	3.34
2^{19}	0.347	0.332e-3	2.83	0.119	0.629e-5	3.29

*Free-boundary spline problem results using the linear stationary iterative process API-DC and the API-CG algorithm. N is the number of unknowns. $per(n) = \| r^n \|_2 / \| r^0 \|_2$. $T(n)$ is the time for n iterations in milliseconds.

6. Conclusion

The concept of an approximate pseudo-inverse provides a useful tool for the implementation of the exact Moore-Penrose pseudo-inverse A^+ of a singular linear operator A and, in some circumstances, allows an $O(N)$ implementation of A^+. We have described a simple construction technique for an API that is useful for consistent singular problems and have applied such APIs to the large sparse underdetermined system arising from free-boundary spline interpolation. The rate of convergence was independent of the order of the problem in this example, which was demonstrated for orders up through a half million, showing that we indeed had an $O(N)$ implementation of A^+. We also demonstrated the effectiveness of the fast approximate pseudo-inverse FAPIN, a V-cycle multigrid algorithm using an API as a smoother, on Poisson's problem on a two-torus of size a quarter million. These APIs proved to be almost perfectly parallelizable, which we demonstrated using the iPSC hypercube at the Christian Michelsen Institute in Bergen. They also proved to be very effective preconditioners for the conjugate gradient algorithm, and allowed the API-CG algorithm to be applied to a nonsymmetric problem, such as the free-boundary spline interpolation problem.

7. Acknowledgements

This work was done in part while M. Benson was visiting the Center for Non-Linear Studies, Los Alamos National Laboratory and while both authors were visiting Christian Michelsen Institute, Bergen, Norway. Support from the U.S. Department of Energy, Norges Teknisk-Naturvitenskapelige Forskningsråd, and the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

REFERENCES

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, John Wiley & Sons, Inc., New York, 1974.

[2] M. W. Benson and P. O. Frederickson, *Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems*, Utilitas Math.,**22** (1982), pp. 127–139.

[3] M. W. Benson, J. Krettmann, and M. Wright, *Parallel algorithms for the solution of certain large sparse linear systems*, Int. J. Comput. Math., **16** (1984), pp. 245-260.

[4] A. Björck and T. Elfving, *Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations*, BIT, **19** (1979), pp. 145–163.

[5] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J.R. Bunch and D.J. Rose, eds., Academic Press, New York, 1976.

[6] T. Elfving, *Block-iterative methods for consistent and inconsistent linear equations*, Numer. Math., **35** (1980), pp. 1–12.

[7] V. Faber and T. A. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., **21** (1984), pp. 352–362.

[8] V. Faber and T. A. Manteuffel, *Orthogonal error methods*, SIAM J. Numer. Anal. (to appear).

[9] P. O. Frederickson and M. W. Benson, *Fast parallel solution of large sparse singular systems*, Chr. Michelsen Institute, Centre for Computer Science report 86/9 (1986).

[10] P. O. Frederickson and M. W. Benson, *Fast parallel algorithms for the Moore-Penrose pseudo-inverse solution to large sparse consistent systems*, Chr. Michelsen Institute, Centre for Computer Science report 86/19 (1986).

[11] W. J. Kammerer and M. Z. Nashed, *On the convergence of the conjugate gradient method for singular linear operator equations*, SIAM J. Numer. Anal., **9** (1972), pp. 165–181.

[12] H. B. Keller, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., **2** (1965), pp. 281–290

[13] O. A. McBryan and E. F. Van de Velde, *Hypercube algorithms and implementations*, Courant Mathematics and Computing Laboratory Report DOE/ER/03077-271 (1986).

[14] E. H. Moore, *On the reciprocal of the general algebraic matrix*, Bull. Amer. Math. Soc., **26** (1919), pp. 394–395.

[15] R. Penrose, *A Generalized Inverse for Matrices*, Proc. Cambridge Philos. Soc., **51** (1955), pp. 406–413.